DRACO7RT · GROUND SYSTEMS · 29 OCTOBER 2026 · SITE NAMES CHECKED 10 NOVEMBER 2026

Ground Systems: Changes to the Baseline Delivery

What the ground segment looked like on 22 July 2026, what was added, changed and fixed since, and what was deliberately left alone

The baseline delivery was one ground station: COSMOS 7.2.1 with the DRACO plugin, logged into directly. Today the same vehicle feeds two equal stations plus a mission planner, behind one sign-in, with retention, a clean-start rule, an archive, operator tooling and a verification script that proves all of it. Every product is still stock; everything that is ours lives in the repository.

Slides 2 and 3: the baseline and the architecture before and after. Slides 4 to 9: COSMOS, access, the viewer and demo control, hygiene, the tee. Slides 10 to 12: Yamcs, PlanDev, Open MCT and Grafana. Slides 13 to 18: operator additions, verification, what was not changed, gaps, timeline, where to find things.

BASELINE

The delivery of 22 July 2026

What the ground segment was

- **COSMOS (OpenC3) 7.2.1**, one station, reached by its own login page
- The base plugin **openc3-cosmos-draco** 1.0.0: the CFS_INT interface (UDP command uplink and telemetry downlink) and the cFE packet database: 20 targets, 19 screens
- TO_LAB aimed straight at COSMOS at 10.5.5.200
- No front door, no second station, no planner, no retention rules, no archive beyond COSMOS's own

What this deck counts as a change

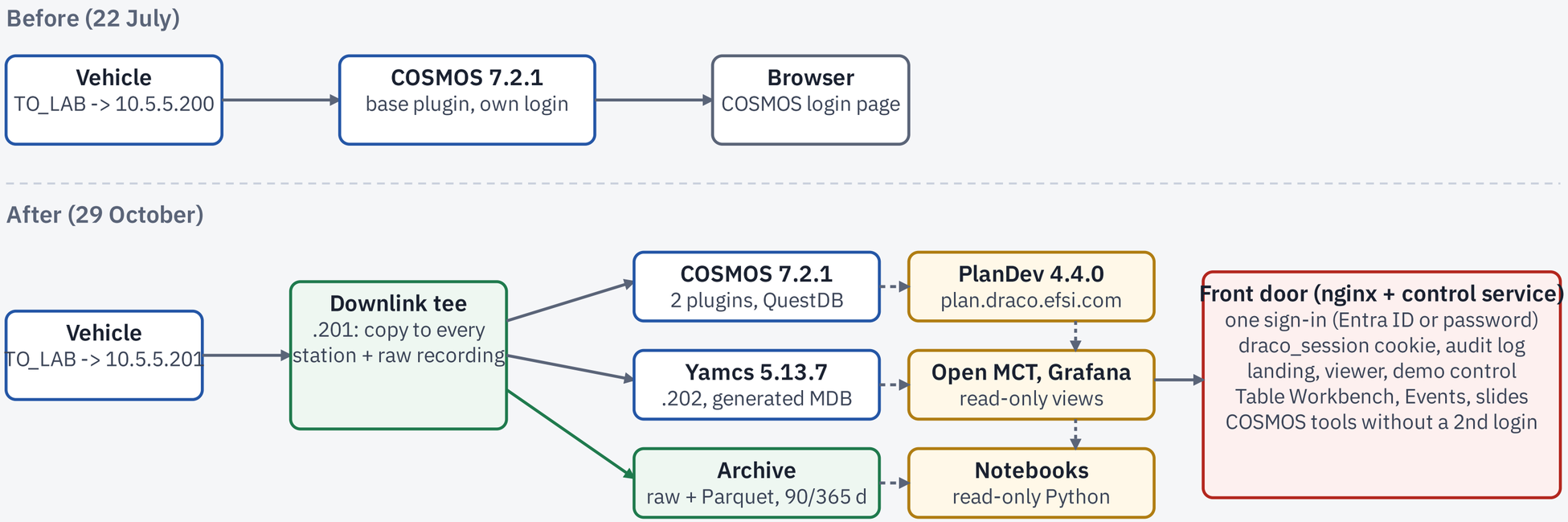
- Anything in the repository under `cosmos/`, `web/`, `run/` and `tools/ground` that did not exist or differs from the baseline
- Fixes to delivered definitions, with their evidence
- Host configuration the ground needs: nginx, systemd units, logrotate, the bind mount

Principles that held throughout

- **COSMOS stays primary and unchanged**: no product code is patched; everything is a plugin, a config or a generated file
- One source of truth for the packet database: the COSMOS definitions. Yamcs's database is generated from them and verification fails on drift
- Everything behind the one sign-in
- Retention from day one for every new store

ARCHITECTURE

Before and after



The vehicle still sends one telemetry stream and still accepts commands on one port; the IPs are baked into its boot image and were not touched. Everything new sits on the fogi_net bridge beside COSMOS or behind nginx on the host.

COSMOS

The two plugins

Base plugin openc3-cosmos-draco, 1.0.0 → 1.0.11

- Delivered: the CFS_INT interface and the cFE packet database (20 targets, 19 screens)
- **Command audit, 22 Oct:** every definition checked against the flight binary's DWARF (sizes, offsets, enums): 257 of 257 commands match on id, function code and length after **28 fixes** (lengths wrong by tail padding, enum values, field offsets). Apps that check length rejected those commands; TO_LAB silently misread them
- **Retention in plugin.txt** (20 Oct): every target keeps packet logs 30 days and QuestDB rows 30 days
- CFDP packets (PDU channels) and the CFDP screen

Upgrade recipe

Bootstrap installs a plugin only when absent. An installed plugin is upgraded by staging the new gem under the *installed* name (`run/upgrade-plugin.sh <dir> <version>`, then front door › Demo control › HEALTH › **Downlink / map**, i.e. --per-boot); installed versions show under Viewer › Tools › **Admin**, Plugins tab; a plain install of a new version crashes COSMOS's installer ("targets:AC already exists") and deletes the old gem.

Mission plugin openc3-fogi-open-source, first commit 21 Aug, now 1.2.4

- 7 targets: ACS, NAV, IMU (the mission apps), GPS42 and IMU42 (what 42 injects on the ground side; never mapped to the vehicle interface), FOGI (the composite pages), SPACECOP (the x86 twin's app)
- 20 screens: 13 FOGI pages (HOME, OVERVIEW, GNC, DEPLOY, PROP, LIMITS, SEQUENCES, EVENTS, APPS, four walls) and the per-app pages
- Two procedures: **fogi_mission_test.rb** (the nine-phase mission test) and **cfdp_demo.rb**
- Derived items, limits on the mission items, saved Telemetry Viewer layouts *wall* and *wall_all* (Viewer › Wall › **Wall (3 screens)** and **All-in-one**)

Two COSMOS 7.2.1 behaviours learned the hard way

- A screen saved from the Telemetry Viewer editor shadows the plugin's copy forever (HOME lost a button that way); delete the modified copy through the screen API
- Never set `tool_log_retain_time`: the cleanup's date parser dies on Script Runner log names and takes the log writer with it

ACCESS

The front door: one sign-in for everything

Delivered

COSMOS's own login page and password, shared by everyone; no record of who did what.

Now (20 Oct, completed 27 Oct)

- Landing page (the front door) at **https://draco.efsi.com/**, top to bottom: header (who is signed in, **Sessions**, **Sign out**), status strip, Ground stations panel, tiles, Demo control
- Signed out, the page shows the card "Mission control sign-in": the blue button **Sign in with Microsoft** (Entra ID, multi-tenant app registration, allowlist of tenants, domains and addresses kept outside git) or the fallback under it: a password box ("Shared COSMOS password") and **Sign in with password**, checked and throttled by the control service (10 Nov; it replaced the link **Sign in to COSMOS**)
- A `draco_session` cookie checked by nginx `auth_request` on every gated path; unsigned visitors bounce to the landing page and back; since 10 Nov that includes COSMOS (pages, API, `/login`)
- **COSMOS tools open without a second login**: the control service mints a COSMOS session per DRACO session with the service password, which never leaves the server
- Also covers **plan.draco.efsi.com** (cookie domain), Grafana, Yamcs, Open MCT, JupyterLab, the viewer, the Table Workbench

Record

- `/var/log/draco-control/audit.log`: every sign-in, sign-out and front-door action with the user (Microsoft email or password:<session>)
- **Sessions** (header, 10 Nov): a Microsoft sign-in can **Sign out** any browser
- `access.log`: every gated request with the user, from the X-Draco-User header auth/check sets
- Kept two years by logrotate

Public on purpose

Only **/slides/** (decks and PDFs) and the sign-in card are open to outside visitors, plus, since 3 Nov, the static file `/yamcs/opi.standalone.js` (Yamcs's open-source display renderer, in a sandboxed frame that gets no cookie; data stays gated). Everything else is gated, COSMOS too since 10 Nov; a new static path needs its own gated location and the exact-match redirect, or it falls through to COSMOS (a 404 there once signed in).

Rule

Absolute *next* URLs after sign-in are allowed only for plan.draco.efsi.com; a sibling host must be added to both the service and the page script.

FRONT DOOR

The viewer and the demo control

The viewer (30 Sep, grown since)

- Opened from the front door tile **Mission Control** (or **Open COSMOS**): https://draco.efsi.com/viewer/. Two button bars on top, the COSMOS page frameless below in a same-origin iframe (CSS hides COSMOS's app bar, drawer and footer; top-right **COSMOS menus: off/on** brings them back, **Open in tab**, **Control** returns to the front door)
- 66 pages (54 on 29 Oct): row 1 = groups **Home**, **Wall**, **GNC**, **Mission**, **cFS** (every core app's housekeeping), **Tools**, **Grafana**, **Yamcs**, **3D view**, and **Plan ↗** (PlanDev in a new tab); row 2 = that group's pages
- **Events dock** along the bottom of every page (29 Oct): **▾ Events** and **Expand** at its left; at its right **Source** (Yamcs/COSMOS), severity chips, a filter box, **Pause**, **Clear**, **Full page**, **Catalog**. Click a row to open the app's screen
- Mission › **Event catalog**, Mission › **Table Workbench**, Tools › **Notebooks (Python)**; since 3 Nov also Mission › **Tutorials** and **Training (step by step)**

Demo control (control service, systemd draco-control)

- The **Demo control** panel at the bottom of the front door: 14 actions, an allowlist run one at a time, in groups STACK (**Start stack**, **Fresh vehicle**, **Shut down**), MISSION (**LAUNCH (RTS-005)**, **Mission test (COSMOS)**), 3D VIEW (**42 start**, **42 stop**), HEALTH (**Verify**, **Clear saved data**, **Downlink / map**), COMMANDING STATION (**Switch to Yamcs**, **Switch to COSMOS**), YAMCS (**Mission test (Yamcs)**, **CFDP file-transfer test**). Each opens a confirm box: **Cancel** / **Run**
- Only the actions that make sense in the current state are enabled; the page greys the rest and explains on hover; the API refuses them unless forced
- Job pane under the buttons: tabs **Job output** (live log) and **Serial console** (read-only UART0, **Pause**, **Clear**), a "recent jobs…" picker at the right
- Status strip at the top, 13 cells: Vehicle, ACS mode, Fuel kg, Deployed, Rate src, IMU, NAV, CFE_ES age, Stack, 42 view, Commanding, Last closed run, Job; under it the **Ground stations** panel (COSMOS, Yamcs, PlanDev, Downlink tee health)

Host side

- systemd `draco-vehicle.service`: after a host reboot the vehicle comes back booted, aimed at the tee, with 42 running (real reboot passed 23 Oct)
- The snap Docker trap: bind mounts work only from $HOME, so the stack runs from `/home/efsi/draco7_0`, a bind mount of the repository

HYGIENE

Retention, clean start and the cleanup timer

The problem (measured 20 Oct)

Unattended, COSMOS wrote 1 to 3 GB a day: QuestDB rows for the 10 Hz 42 streams, raw packet logs (about 90 MB a day) and 1.2 GB a day of “API data” INFO lines from the wall’s polling and the bridge’s inject calls. Yamcs added about 256 MB a day.

Retention (20 Oct)

- Both plugins: packet logs 30 days, QuestDB rows 30 days (existing tables altered by hand)
- Scope text logs 7 days, cleanup poll 10 min
- Vehicle side: compose caps the container json logs; logrotate truncates qemu_debug.log and serial_shell.log at 200 MB
- Yamcs archive 30 days, PlanDev simulation datasets 30 days except the newest per plan

Clean start (28 Oct, project rule)

- A play system must not waste disk: `run/clear-data.sh` wipes COSMOS packet and text logs, every QuestDB table, the Yamcs archive, the cfdpDown bucket, old PlanDev datasets and the vehicle logs on every cold stack start and on front door › Demo control › STACK › **Start stack**; HEALTH › **Clear saved data** runs it alone (always allowed)
- It **keeps** configs, saved screens, Script Runner logs, Yamcs displays, stacks and lists, plans and models, the evidence runs, the audit logs, the shared notebooks and (since 10 Nov) the cfdpUp bucket of files staged to send up
- Since 29 Oct it first **exports every QuestDB table to Parquet** and refuses to truncate if the export fails
- First run freed 55 GB: 12 GB of packet logs, a 51 GB QuestDB volume, 19 million Yamcs rows

Daily cleanup timer (20 Oct, 04:30 UTC)

Prunes Docker build cache older than 7 days and dangling images only (never `image prune -a`: tagged images without a container belong to this and other projects), re-applies the retention settings if lost, archive retention (raw 90 days, Parquet 365 days), reports sizes, warns at 80 % disk.

TRANSPORT

The downlink tee and the raw recorder

Why

TO_LAB sends its one telemetry stream to one address. A second station needs a copy of every datagram, and the vehicle's addresses are baked into its boot image.

What (22 Oct)

- A small container at **10.5.5.201** on fogi_net receives the stream and copies each datagram to every station in `TEE_TARGETS`: COSMOS .200 and Yamcs .202
- Counters per target in /tmp/tee.stats; `--verify` (front door › Demo control › HEALTH › **Verify**) prints them and fails if a station stops receiving while the tee does; the **Downlink tee** card on the front door's Ground stations panel shows the same counters
- A drop-injection file (/tmp/tee.drop) used by the CFDP demo to prove Class 2 recovery at 20 % loss
- Bypass for a direct path: `DOWNLINK_IP=10.5.5.200` on per-boot

Raw recorder (29 Oct)

- The tee appends every datagram to `raw/YYYY-MM-DD.ccsds.gz` with its reception time: the station-neutral record of what the vehicle sent
- About 36 packets a second at about 100 bytes; roughly 310 MB a day before compression; kept 90 days
- Readable from the notebooks (`draco.raw`) and replayable into a station that is not on the live tee

Risk accepted

The tee is a single point for both stations' telemetry. It is kept trivial (one Python file), restarts unless stopped, and the verify script watches its counters.

SECOND STATION

Yamcs 5.13.7, as delivered plus what is ours

Piece	What was added	How it touches Yamcs
Image and instance	Release tarball pinned by sha256, instance *draco*, UDP telemetry from the tee, packets stamped with reception time (the vehicle clock is not UTC), context path /yamcs behind the gate (front door › Ground stations › **Open Yamcs** and Viewer › Yamcs › **Home** open the draco instance's Home page directly)	documented config keys only
Mission database	**Generated** from the COSMOS definitions by `tools/ground/cosmos2xtce.py`: 2,139 parameters, 262 commands, 39 containers; `--check` fails on drift; overlays add Yamcs-only parameters (LC actionpoint states)	XTCE 1.2
Event decoder	`DracoEventDecoder`: the stock cFS decoder assumes a 12-byte telemetry header, this build's is 16; flight events become Yamcs events (source = app, type = EVID)	public Java API
Alarm auto-acknowledge	`DracoAlarmAutoAck`: an alarm leaves the active list once the value is back in limits, stays in history (5.13 cannot configure this)	processor service
Commanding	Link *tc-out* to the vehicle, stock cFS post-processor (length, sequence, XOR checksum), **disabled at start**; switched on the front door, Demo control › COMMANDING STATION › **Switch to Yamcs** / **Switch to COSMOS** (badge shows who commands; strip cell Commanding); one commanding station at a time	stock link class
CFDP engine	The Yamcs CfdpService (entity 21) is **the** CFDP engine for both stations; COSMOS drives it through the REST API; Class 1 and 2 both ways (Yamcs › File transfer; Viewer › Mission › File transfer (CFDP))	stock service
Operator aids	14 displays generated from the COSMOS screens (`cosmos2opi.py`; Yamcs › Telemetry › Displays, folder DRACO), 7 parameter lists (Telemetry › Parameter lists), 9 command stacks (Stacks, folder DRACO), the mission test as a script activity (Scripts, log on Activities; front door › Demo control › YAMCS › **Mission test (Yamcs)**; 9 of 9 in 177 s, 25 Oct)	buckets, REST
Verification	`selfcheck.py` proves every customization is live; part of `--verify`	REST

PLANNING AND VIEWS

PlanDev, Open MCT and Grafana

PlanDev 4.4.0 (NASA-AMMOS, formerly Aerie), 22 Oct

- Own compose project, 11 containers, ports on 127.0.0.1 only, public at **https://plan.draco.efsi.com** behind the same cookie (stock auth is “none”: its own login accepts any name, so the gate is mandatory). Open it from the front door › Ground stations › PlanDev card › **Open Plan**, or **Plan ↗** in the viewer’s top bar (new tab)
- Mission model **DRACO7RT** (Java 21): activities are real commands and RTS (Launch, Deploy, Detumble, MomentumDump, StationKeep, GpsGateCheck, ImuFaultAndRecovery, Downlink); resources follow the ACS source; every constant cited file:line
- **Plan to flight**: a plan becomes a COSMOS Script Runner procedure and is flown on a fresh vehicle; QuestDB actuals come back as an external dataset for plan-versus-actual; plan activities become Grafana annotations

Open MCT 4.3.1, 23 Oct

- Static bundle on the Yamcs plugin (pinned commit; no usable release), served at /openmct/ behind the gate; Viewer › Yamcs › **Open MCT**
- Reads Yamcs through same-origin /yamcs/; tree root DRACO; **read-only**
- A “DRACO starter” layout is created on first visit (/openmct/#reset-starter recreates it); displays live in the browser

Grafana, 30 Sep

- The mission wall and the IMU wall on COSMOS’s QuestDB, provisioned from the repository, kiosk URLs behind the gate: front door tile **Grafana Wall** (/grafana/d/draco-wall) or Viewer › Grafana › **Mission wall** / **IMU wall**
- Dashboards generated by a tool, axes pinned

OPERATOR TOOLING

What operators got that the baseline did not have

Addition	Date	What it does, and where it is
Composite COSMOS pages and walls	Aug to Oct	13 FOGI pages: mission home, GNC, deployments, propulsion, limits and autonomy, sequences, events, app health, four dense 1920×1080 walls, saved layouts. Viewer (tile **Mission Control**) groups Home, Wall, GNC, Mission
Mission test	Aug	Nine phases from pad check to IMU fault recovery, as a Script Runner procedure and, since Oct, as a Yamcs script activity and stack; records evidence runs. Front door › Demo control › MISSION › **Mission test (COSMOS)** and YAMCS › **Mission test (Yamcs)**
Decks and the slides site	30 Sep	Presentations with fixed links (thirteen on 29 Oct, fourteen now), rendered to /slides/ (public) with PDFs. Front door: blue **Presentations** button (header) or tile **Presentations**
CFDP file transfer	Oct	COSMOS CF/CFDP screen (Viewer › Mission › **File transfer (CFDP)**) and demo procedure, Yamcs › File transfer, front-door demo with 20 % loss and evidence (Demo control › YAMCS › **CFDP file-transfer test**)
Table Workbench	27 Oct	RTS editor from the COSMOS dictionary, app-table editors over the compiled layouts, rule checks, delivery by CFDP and Table Services, audited jobs. Tile **Table Workbench** (/tables/), tabs 01 Sequences, 02 App tables, 03 Check, 04 Deliver
Events screen and mute buttons	29 Oct	Mute a chatty app at the source from COSMOS (Viewer › Mission › **Events**: Mute SCH slot chatter, Silence app, …) or a Yamcs stack (9-evs-mute); the Events dock under every viewer page and Mission › **Event catalog**
Notebooks	29 Oct	JupyterLab with the read-only `draco` package over QuestDB, the Yamcs archive, Parquet and the raw record. Tile **Notebooks (Python)** (/lab/) or Viewer › Tools › Notebooks (Python)
Operator notes	Oct	docs/operator-notes: short written procedures beside the decks
Tutorials and Training	3 Nov	Seven hands-on lessons (tile **Tutorials**, /tutorials/) and five step-by-step walkthroughs by role (tile **Training: step by step**, /training/); both also under Viewer › Mission

VERIFICATION

How the ground segment proves itself

bootstrap-draco.sh --verify (no changes; front door › Demo control › HEALTH › Verify)

- Vehicle containers, cFE up, downlink counting, both plugins installed at the expected versions, ACS/NAV/IMU mapped to CFS_INT
- The tee's per-target counters; Yamcs receiving while the tee does; tc-out state
- PlanDev's 11 containers, UI, Hasura health, the public gate
- Since 10 Nov: the gate (COSMOS included), the notebook token limits, no public container ports
- Yamcs `selfcheck.py`: every customization live, displays and stacks uploaded and current

Drift gates

- `cosmos2xtce.py --check`: Yamcs database against COSMOS
- `cosmos2opi.py --check`: generated displays against the screens (hand-edited ones are skipped by hash)
- `evscat.py --check`: event catalog against the flight source
- `tools/tables/catalog.py --roundtrip`: 34 of 34 table layouts byte-identical

Flown, not read

- Mission test from COSMOS and from Yamcs (front door › Demo control › **Mission test (COSMOS)**, **Mission test (Yamcs)**); 9 of 9 phases on both (25 and 26 Oct), including the deliberate FAIL path and a no-42 run
- Closed evidence runs under cFS/evidence/runs with verdicts written back into requirements.csv
- Host reboot test 23 Oct: vehicle back with 42 in 9 minutes, verify clean
- Mute test 29 Oct on both stations (COSMOS: Viewer › Mission › Events buttons; Yamcs: stack 9-evs-mute): event rate 11 to 0 and back

Rule

A pass needs positive evidence the path executed: a counter that moved, a packet timestamp, an event line. The absence of an error is not a pass.

UNCHANGED

What was deliberately left as delivered

Products

- **COSMOS 7.2.1**: no container, image or source changed; configuration lives in the two plugins and the scope settings
- **Yamcs 5.13.7**: release tarball; our two extensions use the public Java API; the mission database is generated
- **PlanDev 4.4.0** and **Open MCT 4.3.1**: stock, pinned; PlanDev's stock auth is why it sits behind the gate
- **Grafana**: stock image, provisioned dashboards

Interfaces

- The vehicle's command port and one telemetry stream (CI_LAB 10.5.5.38:1333, TO_LAB one destination)
- The fogi_net addresses and MACs baked into the U-Boot environment
- The COSMOS command and telemetry definitions as the single source of truth; Yamcs and the tools derive from them

Why it matters

- An upgrade of any product is a version bump and the documented recipe (plugin upgrade, Yamcs upgrade in its README, PlanDev compose)
- Nothing here depends on a fork; a reviewer can diff the repository against the delivery and see every change
- Flight-side changes (cFE Table Services fixes, CF NAK fix, the IMU and table work) are recorded in the flight decks, not here

GAPS

Known gaps and accepted risks

Security

- Yamcs has no security of its own: any signed-in visitor can enable its commanding link from its UI (Yamcs › Links, link tc-out); the rule is procedural
- PlanDev stock auth: any login is admin, so the gate is the only control
- The notebook token is a bearer for reads; it is outside git and recognised only for Yamcs reads and the sql, status, me and health routes; since 10 Nov it cannot get a COSMOS token, so it cannot command

Operations

- One commanding station at a time is not interlocked
- The tee is a single point for both stations' telemetry
- History on the live stores starts at the last stack start; the archive is where older data lives
- COSMOS limits exist on only a few items, so most "went high" questions need a typed threshold

Open items

- SC tables (RTS/ATS) load from the ground waits on a flight fix; app tables work end to end
- The SCH slot events mark real one-second guest stalls with 42 running: cause not chased
- Yamcs walls (the four dense pages) not generated yet
- The COSMOS guide deck still describes the older events route

TIMELINE

When each piece arrived (first commit)

Date	Change
2026-07-22	Baseline: QEMU vehicle, cFS, COSMOS 7.2.1 with the base plugin
2026-08-21	Mission plugin (ACS, NAV, IMU, GPS42, IMU42, FOGI): composite pages, mission test
2026-09-30	Front door: landing page, control service, viewer, decks site, Grafana walls
2026-10-20	Microsoft Entra ID sign-in; retention and the daily cleanup timer; host-reboot recovery
2026-10-22	Downlink tee; Yamcs live with the generated database; PlanDev public; command audit (28 fixes, base plugin 1.0.8)
2026-10-23	Yamcs commanding and events; PlanDev mission model and plan-to-flight loop; Open MCT; Yamcs retention
2026-10-25 to 27	Yamcs displays, operator stacks and script mission test; CFDP; Table Workbench; COSMOS tools without a second login; sign-in record
2026-10-28	Clean start rule: saved data cleared on every stack start (55 GB freed)
2026-10-29	Events dock, catalog, mute buttons on both stations; notebooks; raw recorder; Parquet export; these two decks
2026-11-03	After this deck: training walkthroughs and tutorials (tiles and viewer pages), mission test recorded from both stations, Yamcs display renderer file ungated so displays render
2026-11-10	COSMOS behind the gate; password sign-in on the front door (throttled); **Sessions** sign-out; notebook token read-only; sim.efsi.com redirects; CSRF guard; cfdpUp kept on clear; mission plugin 1.2.4

WHERE

Where to find each change

Change	Repository path	Live
COSMOS plugins	`cosmos/openc3-cosmos-draco, cosmos/openc3-fogi-open-source`	Viewer › Tools › Admin, Plugins tab
Front door, viewer, control service	`web/site, web/viewer, web/control`	https://draco.efsi.com/ (viewer: tile Mission Control, /viewer/)
nginx vhosts (reference copies)	`run/nginx-draco.efsi.com.conf, run/nginx-plan.draco.efsi.com.conf`	/etc/nginx/sites-enabled
Tee, compose, bootstrap, clean start, cleanup	`run/downlink-tee, run/compose.yml, run/bootstrap-draco.sh, run/clear-data.sh, run/disk-cleanup.sh`	docker, systemd timers; front door › Demo control › HEALTH › Verify, Clear saved data
Yamcs	`run/yamcs (README, etc, ext, mdb, displays, operator)`	https://draco.efsi.com/yamcs/ (Open Yamcs: the draco Home page); Viewer › Yamcs
PlanDev, Open MCT, Grafana	`run/plandev, tools/plandev, run/openmct, run/grafana`	plan.draco.efsi.com (Open Plan), /openmct/, /grafana/ (tile Grafana Wall)
Generators and drift checks	`tools/ground (cosmos2xtce, cosmos2opi, evscat), tools/tables, tools/sc`	--check in --verify
Notebooks and archive	`run/jupyter, tools/archive; data in /home/efsi/draco-archive`	https://draco.efsi.com/lab/ (tile Notebooks (Python))
Table Workbench	`web/tables, web/control/tables_api.py, tools/tables, tools/sc`	/tables/ (tile Table Workbench)
Tutorials and Training (3 Nov)	`web/tutorials, web/training, tools/tutorials`	/tutorials/, /training/ (tiles)
Decks	`powerpoint/decks (decks.json)`	/slides/ (public), tile Presentations
Plan and record	`claude-notes/GROUND-PLAN.md, CMD-AUDIT-2026-10-22.md, SECURITY.md, claude-notes.md`	the engineering record

LESSONS

What the ground work taught

Verify against the binary, not the headers

The 28 command fixes came from reading the flight binary's debug information, after months of definitions that *looked* right. Sizes include tail padding; apps that check length reject silently or with one event.

Generate, do not copy

The Yamcs database, the Yamcs displays, the event catalog, the table catalog and the RTS tables are all generated from one source with a drift check. Hand copies would already have diverged.

Retention and a clean start from day one

Every store that was added got its retention the same day. The one that did not (COSMOS, delivered) filled 51 GB before anyone looked.

Keep the products stock

Two Java classes through a public API and a pinned commit are the whole of the product-side footprint. Upgrades stay possible, and the diff against the delivery is the change record itself.

RULES

Rules for the next change to the ground segment

1. **COSMOS stays primary and unchanged.** A ground change is a plugin, a configuration or a generated file, never a patched product.
2. **One source of truth.** The COSMOS definitions; anything derived gets a generator and a `--check` that runs inside `--verify`.
3. **Everything behind the gate.** A new path needs its gated nginx location, the exact-match redirect, and the `auth_request_set` line that names the user in the access log.
4. **Retention and the clean start on the same day** as any new store.
5. **One commanding station at a time.** Yamcs commanding is off unless a procedure turns it on and turns it back off.
6. **Read-only means read-only.** Notebooks, Open MCT and Grafana never get a path to a command.
7. **Prove it live.** A counter that moved, a packet timestamp, a screenshot of the rendered page; then record it in claude-notes.md and, if a convention changed, in CLAUDE.md.
8. **Decks republish to their fixed links.** Fourteen now (thirteen on 29 Oct); the list is powerpoint/decks/decks.json.